

A Guide to NAS-SYS

Listing of Program 2, continued.

```

        0200 ;
        0210 ;
        0220 ;
        0230 ;
        0240 ;Clear screen,move cursor down & right.
2D00 EF      0250     RST PRS
2D01 0C141414 0260     DEFB CLS,CUD,CUD,CUD,CUR,0
1200
        0270 ;
        0280 ;Save current cursor position on stack.
2D07 2A290C 0290     LD HL,(CURSOR)
2D0A E5      0300     PUSH HL
        0310 ;
2D0B CD392D 0320     CALL HLINE ;Top line.
2D0E CD2C2D 0330     CALL VLINE ;Right vertical.
        0340 ;Get the start of current line,
        0350 ; increment it, & move cursor there.
2D11 2A290C 0360     LD HL,(CURSOR)
2D14 DF7C    0370     SCAL CPOS
2D16 23      0380     INC HL
2D17 22290C 0390     LD (CURSOR),HL
        0400 ;
2D1A CD392D 0410     CALL HLINE ;Bottom line.
        0420 ;Get saved cursor position from stack.
2D1D E1      0430     POP HL
2D1E 22290C 0440     LD (CURSOR),HL
        0450 ;
2D21 CD2C2D 0460     CALL VLINE ;Left vertical.
        0470 ;Move cursor to middle of screen,
        0480 ; so that logo will appear there.
2D24 21180A 0490     LD HL,0A18H
2D27 22290C 0500     LD (CURSOR),HL
2D2A DF5B    0510     SCAL MRET
        0520 ;
        0530 ;
        0540 ;
        0550 ;Draw a vertical line from cursor position
2D2C 0605    0560     VLINE LD B,5
2D2E EF      0570     VLL     RST PRS
2D2F 7F11142A 0580     DEFB 7FH,CUL,CUD,2AH,CUL,CUD,0
111400
2D36 10F6    0590     DJNZ VLL
2D38 C9      0600     RET
        0610 ;
        0620 ;Draw a horizontal line from cursor posn.
2D39 0617    0630     HLINE LD B,17H
2D3B EF      0640     H1      RST PRS
2D3C 7F2A00  0650     DEFB 7FH,2AH,0
2D3F 10FA    0660     DJNZ H1
2D41 C9      0670     RET

```

#### 4.3 PROGRAM 3

Program 3 illustrates the use of those routines which output register contents, i.e. B1HEX, B2HEX, TBCD2, TBCD3, and TX1. See Chapter 3 for a full description of each routine.

The output generated by each routine is displayed followed by a dot and newline produced by this program. Routines TBCD2, TBCD3, and TX1 generate checksums in register C and these are also displayed using B2HEX.

TBCD2 outputs 01 and adds this to register C which thus contains 01H. TBCD3 outputs 23H,45H and adds these to register C which therefore contains  $01 + 23 + 45 = 69H$ . TX1 outputs 23H,45H,67H,89H so the checksum becomes  $69 + 23 + 45 + 67 + 89 = 1C1H$ , the overflow 1 being lost.

When run the program outputs:

```
1.  
01.  
01.  
01.  
2345 .  
69.  
2345 6789 .  
C1.  
** NAS-SYS 3 **
```

GD328  
GLW/H

## Listing of Program 3.

```

          0010 ;PROGRAM 3
          0020 ;
2D00      0030      ORG 2D00H
          0040 ;NAS-SYS subroutine numbers.
          0050 ;
2D00 007A  0060 B1HEX EQU 7AH
2D00 0068  0070 B2HEX EQU 68H
2D00 0028  0080 PRS   EQU 28H
2D00 0067  0090 TBCD2 EQU 67H
2D00 0066  0100 TBCD3 EQU 66H
2D00 006C  0110 TX1   EQU 6CH
2D00 005B  0120 MRET  EQU 5BH
          0130 ;
          0140 ;
2D00 EF    0150      RST   PRS
2D01 0C00  0160      DEFB  0CH,0 ;Clear screen.
          0170 ;
2D03 3E01  0180      LD    A,1
2D05 DF7A  0190      SCAL  B1HEX ;Display low nibble of A.
2D07 CD382D 0200      CALL  DOT
          0210 ;
2D0A 3E01  0220      LD    A,1
2D0C DF68  0230      SCAL  B2HEX ;Display A.
2D0E CD382D 0240      CALL  DOT
          0250 ;
2D11 3E01  0260      LD    A,1
2D13 0E00  0270      LD    C,0
2D15 DF67  0280      SCAL  TBCD2 ;Display A, checksum in C.
2D17 CD382D 0290      CALL  DOT
2D1A 79    0300      LD    A,C
2D1B DF68  0310      SCAL  B2HEX ;Display checksum from C.
2D1D CD382D 0320      CALL  DOT
          0330 ;
2D20 214523 0340      LD    HL,2345H
2D23 DF66  0350      SCAL  TBCD3 ;Display HL, checksum in C.
2D25 79    0360      LD    A,C
2D26 DF68  0370      SCAL  B2HEX ;Display checksum from C.
2D28 CD382D 0380      CALL  DOT
          0390 ;
2D2B 118967 0400      LD    DE,6789H
2D2E DF6C  0410      SCAL  TX1   ;Display HL DE, checksum in C.
2D30 79    0420      LD    A,C
2D31 DF68  0430      SCAL  B2HEX ;Display checksum from C.
2D33 CD382D 0440      CALL  DOT
          0450 ;
2D36 DF5B  0460      SCAL  MRET
          0470 ;
          0480 ;Subroutine to display a dot,
          0490 ; followed by a new line.
2D38 EF    0500 DOT   RST   PRS
2D39 2E0D00 0510      DEFB  2EH,ODH,0
2D3C C9    0520      RET

```

#### 4.4 PROGRAM 4

This program shows how arguments may be entered when executing a program.

The program displays a count on the screen; the count may be up or down as determined by the second argument in the command line (0=down count, non-zero = up count), and the step size is determined by the third argument.

The program begins by moving the arguments into register pairs HL, DE, and BC. The up/down argument in DE is examined to see if it is zero or non-zero so determining the direction of the count. The step size in BC (only C is relevant) is negated if a down count is required.

The remainder of the program produces the required count using B2HEX to output the count, TDEL to produce a delay between counts, and CRLF to output each count on a new line.

Sample run:

```
E2D00 1 4
00
04
08
0C
10
14
18
1C
20
```

5/5/28  
W.W.K.

## Listing of Program 4

```

0010 ;PROGRAM 4
0020 ;Up/down counter.
0030 ;Execute as E2D00 <up/down> <step>
0040 ;
2D00      0050      ORG 2D00H
0060 ;
0070 ;NAS-SYS subroutine numbers.
2D00 0060  0080 ARGS EQU 60H
2D00 006A  0090 CRLF EQU 6AH
2D00 0068  0100 B2HEX EQU 68H
2D00 005D  0110 TDEL EQU 5DH
0120 ;
0130 ;
0140 ;
0150 ;
2D00 DF60  0160      SCAL ARGS   ;Move arguments into regs.
2D02 7B    0170      LD A,E     ;If HL
2D03 B2    0180      OR D       ;is not zero,
2D04 2004  0190      JR NZ,GO   ;then upcount, so jump,
2D06 79    0200      LD A,C     ;else
2D07 ED44  0210      NEG        ;negate
2D09 4F    0220      LD C,A     ;the step size.
0230 ;
2D0A AF    0240 GO    XOR A       ;Start count at zero.
2D0B 47    0250      LD B,A     ;B holds current count.
2D0C D9    0260 COUNT  EXX        ;Save active regs.
2D0D DF68  0270      SCAL B2HEX ;Display count from A.
2D0F DF5D  0280      SCAL TDEL   ;Delay.
2D11 DF6A  0290      SCAL CRLF  ;Newline.
2D13 D9    0300      EXX        ;Restore active regs.
2D14 78    0310      LD A,B     ;Add
2D15 81    0320      ADD A,C    ;stepsize from C
2D16 47    0330      LD B,A     ;and put back in B.
2D17 18F3  0340      JR COUNT

```

#### 4.5 PROGRAM 5

This example shows the use of INLIN and RLIN to extract arguments from an input line. B2HEX, TBCD3, and TX1 are used to display the data extracted from the input line.

The program begins in line 240 by calling INLIN to blink the cursor and echo the keyboard input to the video display until Enter is keyed. RLIN then extracts the data from the input line and transfers it to workspace locations ARG1, ARG2, etc, the number of arguments being in ARGN. Lines 260 to 590 display these locations. At line 610, ARG\$ is used to transfer the first three arguments into register pairs HL, DE, and BC. These are also displayed.

Some sample runs:

E2D00  
1234 5678 9ABC DEFO                                ( User input.)  
(ARGN)=04  
(ARG1)=1234  
(ARG2)=5678  
(ARG3)=9ABC  
(ARG4)=DEFO  
1234 5678 9ABC

1 2 3 4 5 6 7 8 9 0                                ( User input.)  
(ARGN)=0A  
(ARG1)=0001  
(ARG2)=0002  
(ARG3)=0003  
(ARG4)=0004  
00001 0002 0003

    ( User input is blank line.)  
(ARGN)=00  
(ARG1)=0001  
(ARG2)=0002  
(ARG3)=0003  
(ARG4)=0004  
0001 0002 0003